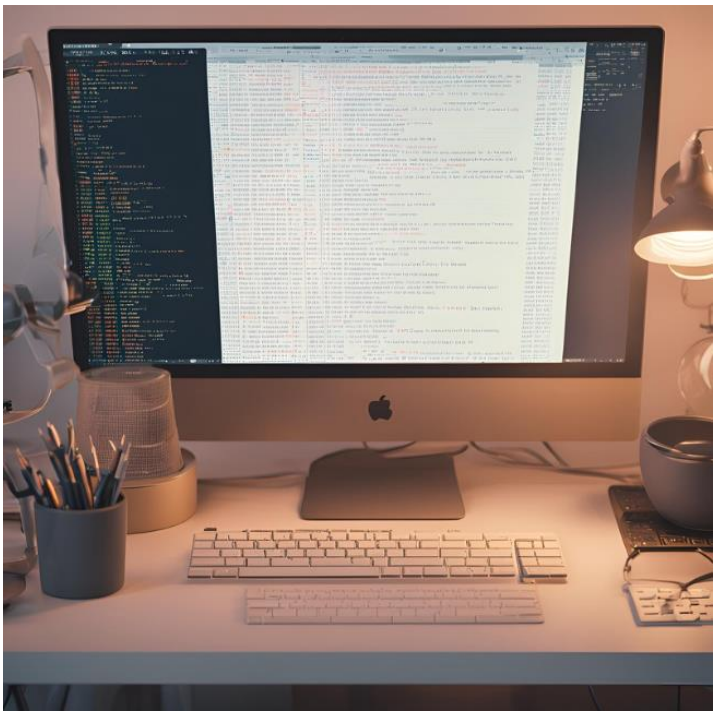
Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

# Ćwiczenie nr 8
# Testy statystyczne w Excelu oraz języku R – część 2



POLITECHNIKA BYDGOSKA
im. Jana i Jędrzeja Śniadeckich

POLITECHNIKA BYDGOSKA
Wydział Technologii i Inżynierii Chemicznej

POLITECHNIKA BYDGOSKA
Wydział Medyczny

PRACOWNIA KOMPUTEROWA

# Wstęp

Najważniejsze elementy języka R potrzebne do obliczeń statystycznych:

## 1. Typy danych i struktury danych w R

R umożliwia łatwe importowanie danych z plików o różnych formatach, takich jak np. CSV, Excel, TXT (funkcje: read.csv(), read.table(), readxl::read_excel()), a także z baz danych SQL i zewnętrznych źródeł internetowych. W przypadku kompilatorów online zwykle dane trzeba podać w postaci tekstu jak opisano powyżej.
Najbardziej elementarnym sposobem przetwarzania danych w R są działania na:

- zmiennych; np. temperatura <- 279, gdzie <- to operator przypisania,
- wektorach; np. wektor_temperatur <- c(279, 300, 298, 310); wektory można też łączyć ze sobą, np. wektor1 <- c(wektor_temperatur, temperatura, wektor_temperatur) stworzy wektor1, który będzie mieć 9 elementów,
- macierzach
  - jeśli macierz jest 2 wymiarowa używamy funkcji matrix();
    np. matrix(1:12, nrow=3), utworzy macierz z wartościami od 1 do 12 w 3-ech rzedach,
    matrix(1:12, ncol=3), utworzy macierz z wartościami od 1 do 12 w 3-ech kolumnach,
    x <- c(1,3,2,10,5)
    y <- 1:5
    m1<-cbind(x,y), złączy wektory po kolumnach i utworzy macierz m1
    m2 <- rbind(x,y), ), złączy wektory po wierszach i utworzy macierz m2
    sięganie do danych: m2[2,3], zwróci element macierzy m2 w 2-gim rzędzie i 3-ciej kolumnie natomiast m2[2,], zwróci cały 2-gi rząd.
  - jeśli macierz ma mieć więcej wymiarów używamy funkcji array()
    np. utworzenie 3-wymiarowej tablicy (array) o wymiarach 2×3×4:
    moje_dane <- 1:24
    moje_array <- array(data = moje_dane, dim = c(2, 3, 4))
    print(moje_array)
- ramkach danych
  - przy analizie danych wykorzystywane są najczęściej ramki danych. Najważniejsze cechy ramki danych: Każda ramka danych powinna zawierać wartości uporządkowane w kolumnach; Każda z kolumn jest wektorem i musi mieć taką samą długość; Różne kolumny mogą przechowywać różne typy danych. Np.:

```
moje_dane <- data.frame(
    imie = c("Anna", "Piotr", "Maria"),
    wiek = c(25, 30, 28),
    miasto = c("Warszawa", "Kraków", "Gdańsk")
)
```

```
   imie wiek   miasto
1  Anna   25 Warszawa
2 Piotr   30   Krakow
3 Maria   28   Gdansk
```

  - niektóre funkcje statystyczne wymagają ramek danych w formacie długim a niektóre w szerokim. **Format długi** jest bardziej elastyczny i często stosowany w analizie powtarzalnych pomiarów, wizualizacji, analizie wariancji i modelach wielopoziomowych. **Format szeroki** jest idealny do porównywania różnych zmiennych, analiz klasteryzacji, korelacji, regresji oraz w modelach dla serii czasowych.
    Np. format **długi** dla ramki moje_dane:

```
library(tidyr)
moje_dane$wiek <- as.character(moje_dane$wiek) # Będziemy chcieli aby wiek i miasto były w kolumnie nazwanej wartość, więc muszą mieć wspólny typ danych, konwertujemy więc wiek na ciąg tekstu
moje_dane_dlugie <- pivot_longer(
  moje_dane,
  cols = c("wiek", "miasto"),  # Kolumny, które mają być przekształcone
  names_to = "atrybut",  # Kolumna na nazwy oryginalnych kolumn
  values_to = "wartosc"  # Kolumna na wartości
)
print(moje_dane_dlugie)
```

W tym przykładzie cols = c("wiek", "miasto") wskazuje, że chcemy przekształcić tylko te kolumny, a imie pozostanie bez zmian.

| | imie | atrybut | wartosc |
|---|---|---|---|
| | <chr> | <chr> | <chr> |
| 1 | Anna | wiek | 25 |
| 2 | Anna | miasto | Warszawa |
| 3 | Piotr | wiek | 30 |
| 4 | Piotr | miasto | Krakow |
| 5 | Maria | wiek | 28 |
| 6 | Maria | miasto | Gdansk |

Jeśli dane są już w formacie długim i chcemy wrócić do formatu **szerokiego**, używamy funkcji pivot_wider():

```
moje_dane_szerokie <- pivot_wider(
  moje_dane_dlugie,
  names_from = atrybut,    # Kolumna, która stanie się nagłówkami kolumn
  values_from = wartosc    # Kolumna zawierająca wartości
)
print(moje_dane_szerokie)
```

| | imie | wiek | miasto |
|---|---|---|---|
| | <chr> | <chr> | <chr> |
| 1 | Anna | 25 | Warszawa |
| 2 | Piotr | 30 | Krakow |
| 3 | Maria | 28 | Gdansk |

## 2. Podstawowe funkcje podsumowujące i statystyczne w R:

- summary() - statystyki podsumowujące
- range() - zwraca jednocześnie minimalną i maksymalną wartość
- rev() - odwraca kolejność elementów
- sort() - sortowanie
- length() - długość obiektu
- mean() - średnia elementów
- sum() - suma elementów
- median() - mediana
- sd() - odchylenie std.

- var() - wariancja
- quantile() - kwartyle, np. quantile(wektor1, probs = c(0.25, 0.5, 0.75))

Każdą z podanych funkcji statystycznych można użyć dla wektorów lub ramek danych podając dla której kolumny ma być coś policzone: np. dla poprzedniego przykładu
moje_dane$wiek <- as.numeric(moje_dane$wiek) #zamieniamy z powrotem znaki na liczbę
quantile(moje_dane$wiek, probs = c(0.25, 0.5, 0.75))

## 3. Testy statystyczne

Testy statystyczne w R pozwalają ocenić istotność zależności, różnic między grupami oraz zgodność z rozkładami.

a) Testy istotności różnic między grupami
   - Test t-Studenta (t.test()) – stosowany do porównania średnich dwóch grup.
   - Test chi-kwadrat (chisq.test()) – stosowany do badania zależności między zmiennymi kategorialnymi; testuje, czy istnieje istotna różnica między obserwowaną a oczekiwaną liczbą przypadków.
   - Analiza wariancji (ANOVA) (aov()) – umożliwia porównanie średnich więcej niż dwóch grup, sprawdzając, czy istnieją istotne różnice między grupami.
   - Analiza przeżycia (survdiff() z pakietu survival) – test log-rank jest powszechnie stosowany w analizach przeżycia do porównania czasów przeżycia między dwiema lub więcej grupami.

b) Testy normalności rozkładu
   - Test Shapiro-Wilka (shapiro.test()) – popularny test sprawdzający, czy rozkład danych nie różni się istotnie od rozkładu normalnego; stosowany dla mniejszych prób (zwykle do 2000 obserwacji).
   - Test Kolmogorova-Smirnova (ks.test()) – również używany do badania zgodności z rozkładem normalnym, ale bardziej odpowiedni dla dużych próbek. Ma możliwość sprawdzenia również innych rozkładów niż normalny.

c) Testy homogeniczności wariancji
   - Test Levene'a (leveneTest() w pakiecie car) – sprawdza jednorodność wariancji między grupami; działa niezależnie od normalności danych.
   - Test F-Fishera (var.test()) – używany do porównania wariancji między dwoma grupami. Jest czuły na normalność, więc najlepiej stosować go dla danych o rozkładzie normalnym.

d) Sprawdzenie mocy testu

## Cel

Celem części drugiej tego ćwiczenia jest praktyczne zastosowanie kolejnych wybranych testów statystycznych w języku R oraz Excelu. Studenci nauczą się, jak:

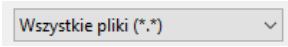
- Wprowadzać i zarządzać danymi w języku R i Excelu.
- Wykonywać podstawowe testy statystyczne, takie jak test t-Studenta, test log-rank, aby sprawdzić zależności i różnice w zestawach danych.
- Interpretować wyniki testów statystycznych.
- Liczyć moc testu.

# Przebieg ćwiczenia

## 1. Testowanie statystyczne danych za pomocą testu t-Studenta – w języku R

Przedmiotem testu są dane ciśnienia skurczowego dla dwóch grup pacjentów (A i B) leczonych innym lekiem na nadciśnienie. Pytanie – czy któryś z leków jest lepszy? Jeśli tak to który?

Przed wykonaniem ćwiczenia zaloguj się na stronę https://posit.cloud i stwórz nowy skrypt np. **Zadanie8_1.R**.

a) Pobranie danych: pobierz plik cisnienie_cw8.csv (link poda prowadzący). Uruchom Excela i otwórz pobrany plik Otwórz->Przeglądaj: aby znaleźć w Excelu plik upewnij się, że ustawiłeś opcję wyświetlania plików na wszystkie: 

Jeśli dane wczytają się w 1 kolumnę popraw to: zaznacz całą kolumnę A i użyj zakładki Dane->Narzędzia danych->Tekst jako kolumny->Rozdzielany (zaznacz przecinek) lub wczytaj dane od nowa: Dane->Pobierz dane->Z pliku CSV. Zadbaj również aby miejsca dziesiętne były oddzielane przecinkami (w języku polskim) Narzędzia główne->Znajdź i zaznacz->Zamień (zamień . na ,). Zapisz plik jako plik Excela (xlsx) – będzie potrzebny jeszcze w kolejnym zadaniu.

b) Przygotowanie danych w języku R: zaznacz obie kolumny w otwartym w Excelu pliku i wkopiuj je w czarne miejsce na stronie: https://tableconvert.com/excel-to-rdataframe lub do **programu z AI** i zamień na format języka R. Poniżej pokazano fragment danych odpowiedni dla tego zadania. Umieść go w nowym skrypcie Zadanie8_1.R.

```
dane <- data.frame(
  grupa = c("A", "A", …),
  wartosci = c("124,9671415", "118,617357", …)
)
```

Następnie zadbaj o odpowiednie przeformatowanie tej ramki danych wpisując:

```
# Zamiana liczb z przecinkiem w formacie znakowym na
odpowiedni do obliczeń (liczba rozdzielana kropka)
dane$wartosci<- as.numeric(gsub(",", ".", dane$wartosci))
# Grupa ma być w formacie czynnik
dane$grupa <- factor(dane$grupa)
# Wyświetlenie ramki danych
print(dane)
```

**Uwaga**: Zmienna typu factor w R jest używana do reprezentowania zmiennych kategorycznych (np. grup, płci, kategorii) i jest bardziej wydajna w analizach statystycznych, ponieważ pozwala na przypisanie etykiet do unikalnych wartości w kolumnie, co może być przydatne w analizach takich jak testy statystyczne, regresja, itp.

c) Uruchom program aby sprawdzić poprawność powyższego kodu. Następnie, załącz wymagane biblioteki i wykonaj test normalności (Shapiro-Wilka) oraz homogeniczności wariancji (Levene):

```
library(ggpubr)     # do wizualizacji wyników
library(dplyr)        # do manipulacji danymi
library(pwr)          # do obliczenia mocy testu
```

```
library(car)      # do Levene

# Filtrujemy dane dla grupy A i B do testu normalności
dane_A <- select(filter(dane, grupa == "A"), wartosci)
dane_B <- selekt(filter(dane, grupa == "B"), wartosci)
# Test Shapiro-Wilka
sA <-shapiro.test(dane_A$wartosci)
sB <-shapiro.test(dane_B$wartosci)
print(sA)
print(sB)
# Test Levene
wynik_levene <- leveneTest(wartosci ~ grupa, data = dane)
print(wynik_levene)
```

d) Wykonaj test t-Studenta aby porównać dane z grupy A oraz B:

```
# Sprawdzenie wyniku testu Levene'a i wybór odpowiedniego
testu t-Studenta
if (wynik_levene$`Pr(>F)`[1] > 0.05) {
  # Równość wariancji - klasyczny test t-Studenta
  wynik_t <- t.test(dane_A, dane_B, var.equal = TRUE)
} else {
  # Różne wariancje - test t-Studenta z korekcją Welcha
  wynik_t <- t.test(dane_A, dane_B, var.equal = FALSE)
}
print(wynik_t)
```

e) Wykonaj sprawdzenie mocy testu

```
library(effsize)
effect_size <- cohen.d(dane_A$wartosci, dane_B$wartosci)
# Moc testu
wynik_mocy <- pwr.t.test(d = effect_size$estimate, n =
length(dane_A$wartosci), sig.level = 0.05, type =
"two.sample")
print(wynik_mocy)
```

f) Zinterpretuj wyniki. Odpowiedz na pytania:
   a. Czy rozkłady A i B są normalne?
   b. Czy wariancje A i B są homogeniczne (jednorodne)?
   c. Czy różnice między A i B są istotne statystycznie? Który lek bardziej pomógł?
   d. Czy moc testu jest wystarczająca? A jeśli nie to co można zrobić żeby była?

g) Narysuj i przeanalizuj wykresy:

```
# Histogramy dla grup A i B
histogram_plot <- ggplot(dane, aes(x = wartosci, fill =
grupa)) +
  geom_histogram(position = "identity", alpha = 0.5, bins =
10, color = "black") +
  labs(title = "Histogramy dla Grup A i B",
```

```
        x = "Wartosci", y = "Częstosc") +
  scale_fill_manual(values = c("lightgreen", "lightblue")) +
  theme_minimal()
print(histogram_plot)
# Histogram z krzywą gęstości
hist_plot <- ggplot(dane_B, aes(x = wartosci)) +
  geom_histogram(aes(y = ..density..), bins = 8, fill =
"lightblue", color = "black") +
  geom_density(color = "red", size = 1) +
  labs(title = "Histogram z Krzywa Gestosci", x =
"Wartosci", y = "Gestosc") +
  theme_minimal()
print(hist_plot)
# Wykres pudełkowy
boxplot_plot <- ggplot(dane, aes(x = grupa, y = wartosci,
fill = grupa)) +
  geom_boxplot() +
  labs(title = "Wykres Skrzynkowy dla Grup A i B",
       x = "Grupa",
       y = "Wartosci") +
  scale_fill_manual(values = c("lightgreen", "lightblue")) +
# Kolory dla grup
  theme_minimal()
print(boxplot_plot)
```

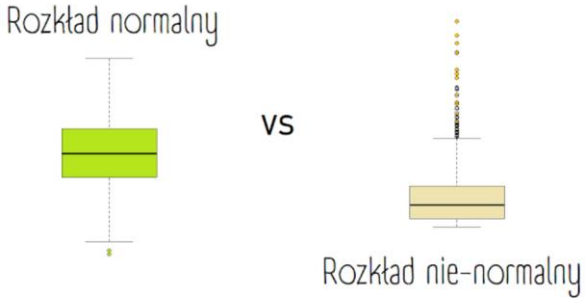
h) Wyeksportuj wykresy do plików png.

i) Zaznacz wykresy oraz skrypt i wyeksportuj je do pliku na dysku (Files->More->Export)

## 2. Testowanie statystyczne danych za pomocą testu t-Studenta – w Excelu

Jeśli zamknąłeś to otwórz ponownie w Excelu plik cisnienie_cw8.xlsx

a) Rozsortuj dane na dwie grupy A oraz B (tabela przestawna lub filtr). Wykonaj statystyki opisowe dla grupy A i B (Dane->Analiza danych->Statystyka opisowa). Wywnioskuj, czy rozkłady danych A i B są normalne (lub zbliżone do normalnych). Możesz to zrobić wizualnie na podstawie histogramu lub wykresu pudełkowego:



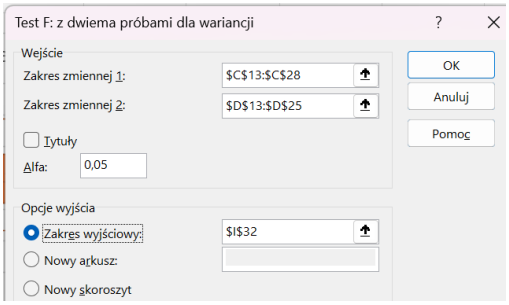

Pudełko zmiennej z rozkładu normalnego jest symetryczne. Może mieć outliery, ale nie może ich być zbyt dużo.

*(Źródło: https://statystykawpsychologii.blogspot.com/2014/08/normalnie-o-normalnym-rozkadzie.html)*

Dodatkowo sprawdź parametry ze statystyki opisowej:

właściwości rozkładu normalnego to jednomodalność, symetryczność (brak skośności), odpowiednia kurtoza. Sprawdź więc skośność i kurtozę: rozkład normalny ma zerową skośność i kurtozę jednakże przez losowość danych nasza skośność i kurtoza nie muszą być 0 aby uznać dane za pasujące do rozkładu normalnego. Jedną z popularniejszych reguł kciuka odnośnie skośności jest ta, która mówi, że skośność w próbie, które znajduje się między -1 a 1 to jest skośność akceptowalna. Dla kurtozy ten przedział wynosi między -2 a 2. Lepszy do tego celu byłby test Shapiro – Wilka lub Kołmogorowa-Smirnova ale nie ma ich standardowo w Excelu (są w Xrealstats).
Jeśli rozkłady są zbliżone do normalnych przejdź do kolejnego punktu.

b) Sprawdź czy wariancje są homogeniczne. Możesz wykorzystać do tego „Test F z dwiema próbami dla wariancji" zawarty w dodatku Analysis ToolPak.
Kliknij Dane->Analiza danych->Test F: z dwiema próbami dla wariancji




gdzie: zakres zmiennej 1 ustaw serie danych ciśnienia dla grupy A, zakres zmiennej 2 ustaw zakres danych dla grupy B, zakres wyjściowy ustaw jedną komórkę gdzie wstawiony ma zostać wynik analizy.
Zinterpretuj wynik: Jeśli wartość statystyki F (pole: test F jednostronny) jest mniejsza niż krytyczna wartość F (pole F), oznacza to, że wariancje są homogeniczne. Możesz też zwrócić uwagę na wartość p. Jeśli jest < 0.05 to odrzucamy hipotezę zerową o homogeniczności wariancji.

c) Wykonaj test t-Studenta klikając Dane->Analiza danych-> Test t: z dwiema próbami zakładający równe/nierówne wariancje (wybór na podstawie poprzedniego wyniku testu F). Spójrz na otrzymane wartości p-value (P(T<=t) jednostronny oraz dwustronny).
d) Wykonaj wykresy histogramu (obie grupy na 1 histogramie: musisz zrobić to ręcznie licząc częstości i wykonać wykresy kolumnowe grupowane, inaczej nie da się uzyskać 2 histogramów na 1 wykresie)
e) Wykonaj wykres pudełkowy i dodaj na nim etykiety.
f) Zinterpretuj wyniki i porównaj je z tymi otrzymanymi w języku R

## 3. Analiza przeżycia przy użyciu testu log-rank – w języku R

Celem ćwiczenia jest zapoznanie się z analizą przeżycia, która pozwala ocenić, czy istnieje statystycznie istotna różnica w czasie przeżycia między dwiema grupami pacjentów.

W wybranej grupie pacjentów porównaj czas przeżycia między pacjentami otrzymującymi standardowe leczenie (Grupa A) a pacjentami otrzymującymi nowe leczenie (Grupa B) w kontekście nowotworów. Pytanie badawcze: czy nowe leczenie wpływa na czas przeżycia pacjentów?

W tym ćwiczeniu wykorzystaj język R oraz pakiety survival i survminer, które oferują narzędzia do analizy przeżycia oraz wizualizacji wyników.

a) **Przygotowanie danych** – przepisz/przekopiuj poniższy kod R, który tworzy przykładowy zbiór 20 danych pacjentów, gdzie:
   - time: czas przeżycia (w miesiącach) dla każdego pacjenta,
   - status: status pacjenta (1 oznacza zgon, 0 oznacza cenzurowanie),
   - group: grupa, do której należy pacjent (A lub B).

```
library(survival)
library(survminer)
survival_data <- data.frame(
  time = c(5, 10, 12, 8, 20, 15, 18, 10, 6, 7, 9, 11, 15,
16, 22, 25, 30, 17, 19, 21),
  status = c(1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1,
0, 1, 1, 1),
  group = c("A", "A", "B", "A", "B", "A", "B", "B", "A",
"A", "B", "B", "A", "B", "A", "B", "A", "B", "A", "B")
)
print(survival_data)
```

b) Stwórz obiekt klasy Surv, który zawiera czas przeżycia (time) oraz status (status). Ten obiekt będzie podstawą do przeprowadzenia analizy przeżycia.

```
surv_obj <- Surv(survival_data$time, survival_data$status)
```

c) Dopasowanie modelu Kaplan-Meier - oblicz krzywe przeżycia dla każdej grupy (A i B) przy użyciu metody Kaplana-Meiera, która umożliwia oszacowanie krzywych przeżycia.

```
fit <- survfit(surv_obj ~ group, data = survival_data)
```

d) Przeprowadź test log-rank, aby sprawdzić, czy istnieje istotna statystycznie różnica między krzywymi przeżycia w grupie A i B. Test log-rank porównuje prawdopodobieństwa przeżycia w poszczególnych grupach w różnych momentach czasowych.

```
log_rank_test <- survdiff(surv_obj ~ group, data =
survival_data)
print(log_rank_test)
```

e) Zinterpretuj wynik testu - wartość $p < 0.05$ wskazuje na istotną różnicę między grupami, co sugeruje, że typ leczenia może wpływać na czas przeżycia pacjentów.

f) Oceń wydajności modelu za pomocą obliczenia zgodności: concordance

```
# Dopasowanie modelu Coxa z uwzględnieniem zmiennej grupy
cox_model <- coxph(surv_obj ~ group, data = survival_data)
summary(cox_model)
```

Uwaga: im zgodność jest bliższa 1 tym lepiej, jeśli jest 0.5 to wskazuje na losowość wyniku analizy.

g) Stwórz wykres krzywych przeżycia dla obu grup, aby lepiej zrozumieć różnice w czasie przeżycia.

```
ggsurvplot(fit, data = survival_data, pval = TRUE, conf.int
= TRUE, title = "Krzywe przezycia dla grup A i B",
legend.title = "Grupa", legend.labs = c("A (Standard)", "B
```

```
(Nowe leczenie)"), xlab = "Czas (miesiace)", ylab =
"Prawdopodobienstwo przezycia")
```

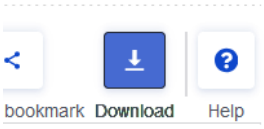
g) Zinterpretuj wyniki:

- Na wykresie przedstawione są dwie krzywe przeżycia dla grup A i B.
- Wartość p z testu log-rank (wyświetlana na wykresie i w wynikach testu) wskazuje, czy istnieje statystycznie istotna różnica między grupami.
- Jeśli $p < 0.05$, możemy wnioskować, że typ leczenia znacząco wpływa na czas przeżycia.
- conf.int = TRUE: powoduje, że na wykresie pojawia się obszar przedstawiający przedział ufności, zazwyczaj na poziomie 95%. Ten obszar daje nam informację o niepewności oszacowania krzywej przeżycia, pozwalając ocenić, jak bardzo model jest pewny swoich prognoz w danym punkcie czasowym. Przedział ufności pokazuje, gdzie krzywa jest mniej precyzyjna (szerszy przedział ufności oznacza większą niepewność). Pozwala to ocenić, czy różnice między grupami są istotne - jeśli przedziały ufności dla dwóch grup nie nakładają się, możemy mieć większą pewność, że różnice między nimi są istotne statystycznie.

## 4. Testowanie statystyczne danych za pomocą testu t-Studenta – zadanie do samodzielnego rozwiązania

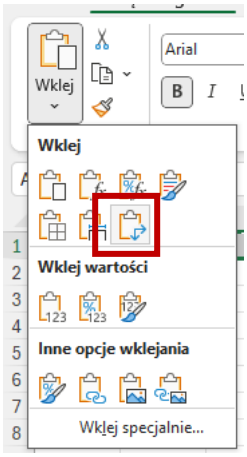
Wejdź na stronę Eurostat https://ec.europa.eu/eurostat/web/health/database
Wejdź do bazy danych długości życia według płci:




Pobierz pełną bazę danych klikając przycisk:
Download->Full dataset->Spreadsheet->Download:
Stwórz nowy plik Excela gdzie będziesz wykonywać analizy a następnie otwórz w Excelu ściągnięty plik. Odkopiuj do swojego pliku dane dla kobiet i mężczyzn w wybranym kraju.

a) Przygotuj dane do analizy tak by mieć 2 kolumny – jedną dla kobiet i drugą dla mężczyzn. Wskazówki: z odkopiowanych linii usuń puste kolumny - zaznacz puste kolumny z przytrzymanym Ctrl a następnie prawy przycisk myszy i „usuń”; potem zaznacz oba wiersze z danymi, kliknij Kopiuj, stań kursorem w komórkę gdzie chcesz wkleić transponowaną tablicę a następnie z menu górnego wybierz Wklej->Transpozycja:




| K | M |
|---|---|
| 81.1 | 72.5 |
| 81.1 | 72.6 |
| 81.2 | 73.0 |
| 81.7 | 73.7 |
| 81.6 | 73.5 |
| 82.0 | 73.9 |
| 81.8 | 73.9 |
| 81.7 | 73.7 |
| 81.9 | 74.1 |
| 80.7 | 72.5 |
| 79.6 | 71.6 |
| 81.1 | 73.4 |

b) Tak jak w poprzednich przykładach wykonaj test normalności i jednorodności wariancji.
c) Wykonaj odpowiedni test t-Studenta.
d) Wykonaj to samo w języku R i dodatkowo sprawdź moc testu.
e) Zinterpretuj wyniki.
f) Dowolnym sposobem wykonaj porównanie 2 grup: albo dla kobiet lub mężczyzn w obrębie 2 różnych krajów; albo w obrębie jednego kraju porównaj czas życia kobiet i mężczyzn.